Probabilidades y Estadística **03 de octubre de 2019**

Tarea 1: Variables, Vectores Aleatorios y Estadística Descriptiva

Profesores: *Gonzalo Fernández, Ewald Stark*

1. Introducción

En la presente tarea, Ud. deberá aplicar sus conocimientos del curso para responder algunas preguntas prácticas. Para ello, deberá modelar un conjunto de problemas, ayudarse de librerías de `Python` que hagan parte del trabajo por usted, y elaborar un informe con análisis de los resultados obtenidos.

El trabajo debe efectuarse en grupos de mínimo 2 personas, máximo 3 personas (en lo posible, que sean de 3 personas). No habrán excepciones fuera de estos números. Por favor no insistir. Tomen en cuenta que este grupo será el mismo para resolver la Tarea 1 y la Tarea 2.

El objetivo de esta tarea es que aprendan a trabajar con una base de datos *DataFrame* en python, para que así puedan interactuar con variables y sus muestras, aplicando los conocimientos vistos en clases e investigando. También especificamente aprender a graficar e interpretar un set de datos para trabajar con ellos y darles interpretaciones de interés.

2. Desarrollo

La tarea está dividida en cuatro partes. Recuerden que cada paso debe venir con comentarios al respecto, principalmente justificando sus decisiones y comentarios de lo que observen y analicen.

2.1. Pregunta 1 (1,5 ptos.)

Resuelva el Control 2 de este semestre usando códigos de `python` referentes a variables aleatorias.

Nota: No está permitido escribir los cálculos de manera textual como código. La pauta ya existe, por ende haga uso de los códigos de python referente a distribuciones.

2.2. Pregunta 2 (1,5 ptos.)

La Gerencia de Marketing de una compañía de telefonía móvil, obtuvo los registros de los minutos consumidos por una muestra de 160 abonados al plan más barato de la empresa, consistente en 250 minutos mensuales como máximo en horas punta. La página Data contiene la lista de los minutos consumidos por cada abonado de la muestra durante un mes. (Utilice la base de datos `data.xlsx` adjunta en la tarea para resolver esta pregunta).

(a) Organice los datos mediante una tabla de distribución de frecuencias.

(b) Grafique los datos mediante un histograma con el polígono de frecuencias asociado.

(c) Calcule e interprete el consumo medio por abonado y su desviación estándar.

(d) ¿Qué porcentaje de abonados se encuentran dentro de una desviación estándar respecto de la media?

Para las Preguntas 3 y 4, utilice la base de datos del Departamento de Estadísticas e Información de Salud (DEIS), en relación especificamente a los nacimientos ocurridos en el país el año 2017. Acceda al sitio web `http://www.deis.cl/bases-de-datos-nacimientos/`. En la página, descargar la información del año 2017 y el Esquema de Registros que explica las variables. Debe importar en python el archivo `.csv` que se descargará del año 2017.

2.3. Pregunta 3 (1,5 ptos.)

(a) Obtenga la esperanza y la matriz de varianza covarianza de las variables cuantitativas. Si omite alguna, justifique el por qué.

(b) Genere un gráfico de dispersión comparando las variables EDAD_M y PESO. ¿Es explicativo? Justifique su respuesta

(c) Genera una matriz de correlaciones considerando las variables cuantitativa que consideró en la parte (a). Existe alguna significativamente alta (o baja)? ¿Porqué ocurre lo anterior? Justifique su respuesta.

2.4. Pregunta 4 (1,5 ptos.)

(a) Genere el gráfico adecuado para observar la cantidad de nacimientos según estado civil de la madre.

(b) Genere un histograma para la cantidad de nacimientos según intervalos de pesos de los hijos. Para ello, defina un intervalo de pesos de los hijos según lo visto en clases.

(c) Genere un diagrama de caja (boxplot) para los datos de la parte (b), es decir, un diagrama de caja para los intervalos creados para el peso de los niños y sus cantidades. Además, genere otros 3 diagramas de cajas (en un mismo gráfico) que explique los percentiles de los pesos anteriores en la Región Metropolitana, otro para la Región de Antofagasta y otro para la Región de Los Lagos.

(d) Genere un gráfico adecuado para explicar la cantidad de nacimientos en cada mes del año. ¿Existe alguna tendencia? Justifique.

3. Sobre la entrega

En un archivo comprimidio en `zip` deberá entregar:

- Un informe en `pdf` con portada y hasta diez páginas de contenido (incluyendo figuras). En la portada debe explicitarse el nombre completo y RUT de cada uno de los integrantes del grupo.
- El código en `Python` utilizado en cada pregunta. El nombre de archivo deberá tener de la forma: `P` + *nro. de pregunta* + `.py` (p.ej., `P2.py`).

El archivo comprimido deberá tener por nombre `Tarea1_` + *Primer apellido de cada integrante del grupo* + `.zip`. P.ej., `Tarea3_SanchezVidalBravo.zip`. El incumplimiento de estas restricciones de nombres será sancionado con un descuento de un punto en la nota final de la tarea.

El archivo comprimido deberá entregarse a través del buzón habilitado para ese fin en la página del curso en SAF. El plazo para entregar vence impostergablemente el **Viernes 18 de Octubre a las 23:59 hrs.**

4. Librerías de Python

En esta sección se entrega una ayuda introductoria para el uso de dos librerías que necesitará para esta tarea: `itertools` y `scipy`. Finalmente se explica cómo instalarlas, si no están disponibles en el computador en que Ud. va a trabajar. Se asume que `Python` sí está instalado, junto con algún entorno de programación.

Si necesita más ayuda, recuerde que en internet puede encontrar tutoriales de ambas librerías y de `Python` en general para todos los niveles de dificultad, y que también puede preguntar a los profesores y ayudantes del curso.

4.1. `pandas` para trabajar bases de datos

Necesitaran de esta libreria para trabajar de manera más ágil y fácil con las bases de datos. Su uso es:

```
import pandas as pd
```

Les recomiendo ver bien estos tres enlaces bien completos de como utilizar `pandas` y los llamados `DataFrame` para tratar bases de datos.

- `https://datacarpentry.org/python-ecology-lesson-es/03-index-slice-subset/index.html` Este ve en particular lo básico de `DataFrame` en español, y como cargar la base de datos.
- `https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python` `pandas`
- `https://ourcodingclub.github.io/2018/04/18/pandas-python-intro.html` `pandas` muy completo y con gráficos. Les servirá mucho para los gráficos que tienen que hacer

4.2. para cargar la base de datos

Una vez que importen `pandas`, deberán escribir el siguiente código:

```
bdd = pd.read_csv("ubicaci ndelarchivo")
```

4.3. GRÁFICOS

Usen la página `https://python-graph-gallery.com/`, es excelente y muy completa para todo lo que es gráficos. Si tienen dudas en como interactuar con su base de datos y gráficos, pueden servirles también las siguientes páginas:

- `https://relopezbriega.github.io/blog/2016/09/18/visualizaciones-de-datos-con-python/` .
- `https://towardsdatascience.com/a-guide-to-pandas-and-matplotlib-for-data-exploration-56fad95f951`
- `http://queirozf.com/entries/pandas-dataframe-plot-examples-with-matplotlib-pyplot`

4.4. `scipy.stats` para variables aleatorias

La librería `scipy.stats` permite manejar distribuciones aleatorias y generar variables según esas distribuciones. Para usarla, debe importarla. Por ejemplo:

```
import scipy.stats as st
```

Si este paso arroja un mensaje de error, vea la sección 4.6.

En general, en esta librería las distribuciones pueden ser tratadas de manera análoga. A continuación se muestra algunos ejemplos con la distribución $Beta(2, 2)$.

```
#crear una distribucion Beta(2,2)
d=st.beta(2,2)

#generar 10 numeros aleatorios segun esa distribucion
#y mostrarlos en la consola
v=d.rvs(10)
print(v)

#evaluar la funcion de densidad de probabilidad
print(d.pdf(0.5))
```

```
#evaluar la funcion de probabilidad acumulada
print(d.cdf(0.5))
```

El uso de los métodos `rvs` para generar valores a partir de una distribución, así como `pdf` y `cdf` para las funciones de densidad respectivas (f_X y F_X), se aplica a todas las distribuciones disponibles en `scipy.stats`.

Las siguientes son algunas de las distribuciones disponibles:

Nombre castellano	Función `scipy`
Beta	`beta`
Chi-cuadrado/χ^2	`chi2`
Exponencial	`expon`
Gamma	`gamma`
Log-Normal	`lognorm`
Logística	`logistic`
Normal	`norm`
Uniforme	`uniform`

4.5. `matplotlib.pyplot` para graficar

Si lo juzga conveniente, puede investigar sobre la librería `matplotlib` de `Python` para graficar.

La librería `matplotlib.pyplot` permite hacer gráficos con pocas líneas de código. Para usarla, debe importarla. Por ejemplo:

```
import matplotlib.pyplot as pt
```

Para crear una figura en blanco se usa `figure`:

```
pt.figure()
```

El comando `hist` crea histogramas automáticamente a partir de una lista de datos:

```
d=st.beta(4,4) #esta linea requiere import scipy.stats as st
pt.hist(d.rvs(1000), normed=True)
```

... la opción `normed=True` *normaliza* el histograma, escalándolo de manera que cubra un área total de 1, igual que una función de densidad de probabilidad.

El comando `plot` dibuja líneas dados una lista de coordenadas x y otra de coordenadas y (la línea será la poligonal que une los puntos de la forma (x_i, y_i)):

```
d=st.beta(4,4) #esta linea requiere import scipy.stats as st
x=[0.01*i for i in range(100)]
y=[d.pdf(xi) for xi in x]
pt.plot(x,y)
```

4.6. Instalación de librerías

Si una librería no está instalada junto a `Python` en un computador dado, la consola entregará un mensaje de error de la forma `ImportError: No module named 'pepito'`. Para instalarla, escriba las siguientes dos líneas de código, reemplazando "pepito" por el nombre de la librería, entre comillas:

```
import pip
pip.main(["install", "pepito"])
```